UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/826,723	04/16/2004	Andreas S. Krebs	11884/502501	1351

53000 7590 11/14/2007

KENYON & KENYON LLP
1500 K STREET N.W.
WASHINGTON, DC 20005

EXAMINER	
ABDUL-ALI, OMAR R	
ART UNIT	**PAPER NUMBER**
2178	

MAIL DATE	DELIVERY MODE
11/14/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/826,723	KREBS, ANDREAS S.	
	Examiner	Art Unit	
	Omar Abdul-Ali	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 September 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17 and 19-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17, and 19-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

The following action is in response to the response filed September 5, 2007. Amended Claims 1-17, and 19-35 are pending and have been considered below.

1. Examiner's Note: The prior art rejections have been withdrawn as necessitated by Applicant's amendments.

2. Examiner's Note: The 35 U.S.C. 112 second paragraph rejection has been withdrawn as necessitated by Applicant's amendment.

3. Examiner's Note: The 35 U.S.C. 101 rejections have been withdrawn as necessitated by Applicant's amendment.

Claim Objections

4. Claim 24 is objected to because of the following informalities: Claim 24 depends on previously cancelled Claim 18. Appropriate correction is required.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-6, 8-14, 16, 17, 19-22, 24-30, and 32-35 rejected under 35 U.S.C. 103(a) as being unpatentable over Oni (US 2004/0133546) in view of Emmerichs (US 2003/0061482).

Claim 1: Oni discloses a method of managing visibility of GUI components in an application, comprising:

a. initializing the application (page 3, paragraph 43);

Oni discloses invoking a visibility manager for a plurality of profiles and displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager (page 4, paragraph 62), but does not explicitly disclose providing a user interface of the visibility manager for selection of one of a plurality of visibility states for each of at least a subset of the GUI components. Emmerichs discloses a similar method of managing visibility of GUI components in an application that further discloses selecting options in a security manager (visibility manager) that control the visibility states of GUI objects accessed by a user through a user interface. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a user interface of the visibility manager in Oni, because providing a selection interface was recognized as part of the ordinary capabilities of one skilled in the art. One would have been motivated to provide a user interface of the visibility manager in order to provide access to GUI components based on access privileges

Claim 2: Oni and Emmerichs disclose a method of managing visibility of GUI components in an application as in Claim 1 above, and Oni further discloses:

a. reading the plurality of profiles (page 4, paragraphs 63 and 67);

b. processing the plurality of profiles (page 4, paragraphs 63 and 67);

c. reading and processing a user configuration based on the plurality of profiles (page 4, paragraphs 62 and 67);

d. activating the applied profile (page 4, paragraph 65).

Claim 3: Oni and Emmerichs disclose a method of managing visibility of GUI components in an application as in Claim 2 above, and Oni further discloses:

a. revising the user interface based on the applied profile (page 4, paragraph 65);

b. displaying the user interface (page 4, paragraph 65);

Claim 4: Oni and Emmerichs disclose a method of managing visibility of GUI components in an application as in Claim 3 above, and Oni further discloses:

a. starting the application (page 3, paragraph 43);

b. building the user interface with all the GUI components visible (page 4, paragraph 62);

c. calling the visibility manager after the applied profile is activated (page 4, paragraph 63)

Claim 5: Oni and Emmerichs disclose a method of managing visibility of GUI components in an application as in Claim 2 above, and Oni further discloses:

a. selecting an identification of a particular GUI component (page 4, paragraph 67);

b. locating the identification in a mapping table [dynamic repository] (page 4, paragraph 67);

c. checking a state of the particular GUI component (page 4, paragraph 69);

d. comparing the state to the applied profile (page 4, paragraph 69);

e. changing the state if not in agreement with the applied profile (page 4, paragraph 69);

f. repeating locating the identification, checking the state, comparing the state, and changing the state for any remaining identifications of additional GUI components (page 4, paragraph 69).

Claim 6: Oni and Emmerichs disclose a method of managing visibility of GUI components in an application as in Claim 5 above, and Oni further discloses:

a. the state is visible or not visible (page 4, paragraph 69).

Claim 8: Oni and Emmerichs disclose a method of managing visibility of GUI components in an application as in Claim 1 above, and Oni further discloses:

a. reading the plurality of profiles (page 4, paragraph 63);

b. processing the plurality of profiles (page 4, paragraph 63);

c. reading and processing a user configuration based on the plurality of profiles (page 4, paragraph 62);

d. activating the applied profile by:

1) selecting an identification of a particular GUI component (page 4, paragraph 67);

2) locating the identification in a mapping table (page 4, paragraph 67);

3) checking a state of the particular GUI component (page 4, paragraph 69);

4) changing the state if not in agreement with the applied profile (page 4, paragraph 69);

5) repeating locating the identification, checking the state, comparing the state, and changing the state for any remaining identifications of additional GUI components (page 4, paragraph 69).

e. revising the user interface based on the applied profile (page 4, paragraph 65);

f. displaying the user interface (page 4, paragraph 65);

g. initializing the application by:

1) starting the application (page 3, paragraph 43);

2) building the user interface with all of the GUI components visible (page 4, paragraphs 62-63);

3) calling the visibility manager after the applied profile is activated (page 4, paragraphs 62-63).

Claim 9: Oni discloses a system of managing visibility of GUI components in an application, comprising:

a. a processor (page 4, paragraph 68). Oni discloses a computer system implementation, and it is inherent that a processor is present.

a. a user interface module of the application, including GUI components (page 4, paragraph 65);

b. a visibility manager that determines which GUI components are visible (page 4, paragraph 62).

Oni discloses invoking a visibility manager for a plurality of profiles and displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager (page 4, paragraph 62), but does not explicitly disclose providing a user interface of the visibility manager for selection of one of a plurality of visibility states for each of at least a subset of the GUI components. Emmerichs discloses a similar method of managing visibility of GUI components in an application that further discloses selecting options in a security manager (visibility manager) that control the visibility states of GUI objects accessed by a user through a user interface. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a user interface of the visibility manager in Oni, because providing a selection interface was recognized as part of the ordinary capabilities of one skilled in the art. One would have been motivated to provide

a user interface of the visibility manager in order to provide access to GUI components based on access privileges.

Claim 10: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 9 above, and Oni further discloses:

a. read the one or more profiles upon initialization of the application (page 4, paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process a user configuration based on the one or more profiles (page 4, paragraph 62);

d. activate a particular profile of the one or more profiles (page 4, paragraph 65).

Claim 11: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 10 above, and Oni further discloses:

a. build the user interface with all the GUI components visible (page 4, paragraph 62);

b. call the visibility manager after the particular profile of the one or more profiles is activated (page 4, paragraph 63);

c. revise the user interface based on the particular profile of the one or more profiles (page 4, paragraph 65);

d. display the user interface (page 4, paragraph 65).

Claim 12: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 10 above, and Oni further discloses:

a. select an identification of a particular GUI component (page 4, paragraph 67);

b. locate the identification in a mapping table (page 4, paragraph 67);

c. check a state of the particular GUI component (page 4, paragraph 69);

d. compare the state to the particular profile of the one or more profiles (page 4, paragraph 69);

e. change the state if not in agreement with the particular profile of the one or more profiles (page 4, paragraph 69);

Claim 13: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 12 above, and Oni further discloses:

a. activating the particular profile of the one or more profiles further comprises repeating locating the identification, checking the state, comparing the state, and changing the state for any remaining identifications of additional GUI components (page 4, paragraph 69).

Claim 14: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 12 above, and Oni further discloses:

a. the state is visible or not visible (page 4, paragraph 69).

Claim 16: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 9 above, and Oni further discloses:

a. read the one or more profiles upon initialization of the application (page 4, paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process a user configuration based on the one or more profiles (page 4, paragraph 62);

d. select an identification of a particular GUI component (page 4, paragraph 67);

e. locate the identification in a mapping table (page 4, paragraph 67);

f. check a state of the particular GUI component (page 4, paragraph 69);

g. compare the state to the particular profile of the one or more profiles (page 4, paragraph 69);

h. change the state if not in agreement with the particular profile of the one or more profiles (page 4, paragraph 69);

i. start the application (page 3, paragraph 43);

j. build the user interface with all the GUI components visible (page 4, paragraph 62);

k. call the visibility manager after the particular profile of the one or more profiles is activated (page 4, paragraph 63);

l. revise the user interface based on the particular profile of the one or more profiles (page 4, paragraph 65);

m. display the user interface (page 4, paragraph 65);

Claim 17: Oni discloses a system of managing visibility of GUI components in an application, comprising:

a. a processor;

b. use a visibility manager data structure in managing visibility of GUI components in a user interface of an application, the visibility manager data structure comprising a mapping table, one or more profiles and a user configuration identifying which of the one or more profiles is to be applied (page 4, paragraphs 65 and 67). Specifically, Oni discloses linking user interface elements with the repository of user profiles, and matching the profile of a new user with the profile of previously stored profiles for which combinations of user interface elements exist in the solution repository.

Oni discloses invoking a visibility manager for a plurality of profiles and displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager (page 4, paragraph 62), but does not explicitly disclose providing a user interface of the visibility manager for selection of one of a plurality of visibility states for each of at least a subset of the GUI components. Emmerichs discloses a similar method of managing visibility of GUI components in an application that further discloses selecting options in a security manager (visibility manager) that control the visibility states of GUI objects accessed by a user through a user interface. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a user interface of the visibility

manager in Oni, because providing a selection interface was recognized as part of the ordinary capabilities of one skilled in the art. One would have been motivated to provide a user interface of the visibility manager in order to provide access to GUI components based on access privileges

Claim 19: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 17 above, and Oni further discloses:

a. read the one or more profiles upon initialization of the application (page 4, paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process the user configuration based on the one or more profiles (page 4, paragraph 62);

d. activate the identified profile of the one or more profiles based upon the mapping table (page 4, paragraph 67);

Claim 20: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 19 above, and Oni further discloses:

a. revise the user interface based on the identified profile of the one or more profiles (page 4, paragraph 65);

b. display a user interface (page 4, paragraph 69);

Claim 21: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 19 above, and Oni further discloses:

a. select an identification of a particular GUI component (page 4, paragraph 67);

b. locate the identification in a mapping table (page 4, paragraph 67);

c. check a state of a specific GUI component in the application (page 4, paragraph 69);

d. compare the state to the particular profile of the one or more profiles (page 4, paragraph 69)

e. change the state if not in agreement with the identified profile of the one or more profiles (page 4, paragraph 69);

Claim 22: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 21 above, and Oni further discloses:

a. the state is visible or not visible (page 4, paragraph 69).

Claim 24: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 19 above, and Oni further discloses:

a. read the one or more profiles upon initialization of the application (page 4, paragraph 63);

b. process the one or more profiles (page 4, paragraph 63);

c. read and process a user configuration based on the one or more profiles (page 4, paragraph 62);

d. select an identification of a particular GUI component (page 4, paragraph 67);

e. locate the identification in the mapping table (page 4, paragraph 67);

f. check a state of the particular GUI component (page 4, paragraph 69);

g. compare the state to the identified profile of the one or more profiles (page 4, paragraph 69);

h. change the state if not in agreement with the identified profile of the one or more profiles (page 4, paragraph 69);

i. revise the user interface based on the activated profile (page 4, paragraph 65);

j. display a user interface (page 4, paragraph 69).

Claim 25: Oni discloses a system of managing visibility of GUI components in an application, comprising:

a. initializing an application (page 3, paragraph 43);

b. invoking a visibility manager (page 4, paragraph 62);

Oni discloses invoking a visibility manager for a plurality of profiles and displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager (page 4, paragraph 62), but does not explicitly disclose providing a user interface of the visibility manager for selection of one of a plurality of visibility states for each of at least a subset of the GUI components. Emmerichs discloses a similar method of managing visibility of GUI components in an application that further discloses selecting options in a security manager (visibility manager) that control the visibility states of GUI objects accessed by a user through a

user interface. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a user interface of the visibility manager in Oni, because providing a selection interface was recognized as part of the ordinary capabilities of one skilled in the art. One would have been motivated to provide a user interface of the visibility manager in order to provide access to GUI components based on access privileges

Claim 26: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 25 above, and Oni further discloses:

a. reading the one or more profiles (page 4, paragraph 63);

b. processing the one or more profiles (page 4, paragraph 63);

c. reading and processing a user configuration based on the one or more profiles (page 4, paragraph 62);

d. activating a particular profile of the one or more profiles (page 4, paragraph 67);

e. revising the user interface based on the particular profile of the one or more profiles (page 4, paragraph 65);

f. displaying the user interface (page 4, paragraph 69).

Claim 27: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 26 above, and Oni further discloses:

a. starting the application (page 3, paragraph 43);

b. building the user interface with all of the GUI components visible (page 4, paragraph 62);

c. calling the visibility manager after the particular profile of the one or more profiles is activated (page 4, paragraph 63);

Claim 28: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 26 above, and Oni further discloses:

a. selecting an identification of a particular GUI component (page 4, paragraph 67);

b. locating the identification in a mapping table (page 4, paragraph 67);

c. checking a state of the particular GUI component (page 4, paragraph 69);

d. comparing the state to the particular profile of the one or more profiles (page 4, paragraph 69);

e. changing the state if not in agreement with the particular profile of the one or more profiles (page 4, paragraph 69);

Claim 29: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 26 above, and Oni further discloses:

a. repeating locating the identification, checking the state, comparing the state, and changing the state for any remaining identifications of additional GUI components (page 4, paragraph 69).

Claim 30: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 26 above, and Oni further discloses:

a. the state is visible or not visible (page 4, paragraph 69).

Claim 32: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 25 above, and Oni further discloses:

a. reading the one or more profiles (page 4, paragraph 63);

b. processing the one or more profiles (page 4, paragraph 63);

c. reading and processing a user configuration based on the one or more profiles (page 4, paragraph 62);

d. selecting an identification of a particular GUI component (page 4, paragraph 67);

e. locating the identification in a mapping table (page 4, paragraph 67);

f. checking a state of the particular GUI component (page 4, paragraph 69);

g. repeating locating the identification, checking the state, comparing the state, and changing the state for any remaining identifications of additional GUI components (page 4, paragraph 69).

h. revising the user interface based on the particular profile of the one or more profiles (page 4, paragraph 65);

i. displaying the user interface (page 4, paragraph 69);

j. starting the application (page 3, paragraph 43);

k. building the user interface with all of the GUI components visible (page 4, paragraph 62);

l. calling the visibility manager after the profile is activated (page 4, paragraph 63);

Claim 33: Oni discloses a system for managing visibility of GUI components in an application, comprising:

a. first means for interfacing with a user, the means for interfacing including GUI components for display (page 4, paragraph 62);

Oni discloses means for determining which GUI components are visible (page 4, paragraph 62), but does not explicitly disclose the means for determining including second means for interfacing with a user, the second means for interfacing providing for receipt, for one or more profiles, of respective user selections of visibility states of at least a subset of the GUI components, the selections used for the determination of which GUI components are visible. Emmerichs discloses a similar method of managing visibility of GUI components in an application that further discloses selecting options in a security manager (visibility manager) that control the visibility states of GUI objects accessed by a user through a user interface. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a user interface of the visibility manager in Oni, because providing a selection interface was recognized as part of the ordinary capabilities of one skilled in the art. One would have

been motivated to provide a user interface of the visibility manager in order to provide access to GUI components based on access privileges

Claim 34: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 33 above, and Oni further discloses:

a. means for reading the one or more profiles upon initialization of the application (page 4, paragraph 63);

b. means for processing the one or more profiles (page 4, paragraph 63);

c. means for reading and processing a user configuration based on the one or more profiles (page 4, paragraph 62);

d. means for activating a particular profile of the one or more profiles (page 4, paragraph 67).

Claim 35: Oni and Emmerichs disclose a system of managing visibility of GUI components in an application as in Claim 34 above, and Oni further discloses:

a. means for starting the application (page 3, paragraph 43);

b. the first means for interfacing with the user initially build the user interface with all the GUI components visible (page 4, paragraph 62);

c. the means for determining which GUI components are visible performs the determination after the particular profile of the one of more profiles is activated (page 4, paragraph 62);

d. the first means for interfacing with the user displays the user interface (page 4, paragraph 69).

6. Claims 7, 15, 23, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Oni (US 2004/0133546) in view of Emmerichs (US 2003/0061482) and further in view of Moshfeghi (US 6,476,833).

Claims 7, 15, 23, and 31: Oni discloses a method, system, data structure, and medium embodying instructions as in Claims 5, 12, 21, and 28 above, further comprising the mapping table comprises a plurality of identifications of GUI components (page 4, paragraph 67). However, Oni does not explicitly disclose a corresponding plurality of objects of an object-oriented and platform independent programming language. Moshfeghi discloses a similar method, system, data structure, and medium embodying instructions that further discloses the GUI components of the application are specified with the Swing component set of the Java™ Foundation Classes (Column 15, lines 19-31). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made that the mapping table could comprise identifications of references to objects of an object-oriented and platform independent programming language. One would have been motivated to include Java software objects in view of the fact that Java is a widely used programming language throughout the Internet and World Wide Web (WWW).

Response to Arguments

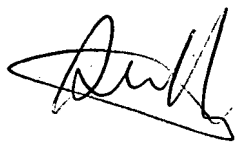
7. Applicant's arguments with respect to claims 1-17, and 19-35 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Omar Abdul-Ali whose telephone number is 571-270-1694. The examiner can normally be reached on Mon-Fri(Alternate Fridays Off) 8:30 - 6:00 EST.



STEPHEN HONG
SUPERVISORY PATENT EXAMINER